Lab 3: Keypad Scanner

Introduction

In this lab you will learn how to use an FPGA to scan inputs from a keypad.

Learning Objectives

By the end of this lab you will have…

- Designed a circuit interface to read a matrix keypad.
- Understood and implemented a solution to deal with switch bouncing.
- Implemented robust digital design strategies to ensure appropriate behavior with a variety of different user inputs.

Requirements

Design and construct a circuit on your FPGA to read a 4-by-4 matrix keypad. Display the last two hexadecimal digits pressed on your dual seven-segment display, with the most recent entry appearing at the right. The keypad and your display should be in the same orientation.

Each key press should be recorded exactly once, at the time the key is pressed, no matter how long or short a time the user holds the key down. If additional keys are pressed while still holding down the first key, these inputs should be ignored. The seven-segment displays should not flicker and should be the same brightness regardless of how many segments are displayed.

Open specifications in new tab.

Lab 3 Specifications

Lab-specific Specifications

Proficiency

- ☐ Circuit correctly reads inputs from 4×4 keypad.
- ☐ Dual seven-segment display shows the last two hexadecimal digits pressed.
- ☐ Most recent numeric entry is shown on the right.
- ☐ Design does not lock up when multiple buttons are pressed at once. (i.e., it just holds the current values on the display and functions properly again when the buttons are released.)
- ☐ Design only registers first button press if additional buttons are pressed down while holding down one button.
- ☐ Each button press registered only once (e.g., no switch bouncing)
- ☐ Seven segment displays are same brightness regardless of how many segments are illuminated.
- ☐ Design has no latches.
- ☐ Design has no tristate buffers.
- ☐ Report includes state transition diagram illustrating the operation of the system.

Excellence

- ☐ Design uses synchronizers on asynchronous inputs to mitigate metastability.
- ☐ Keypad and seven-segment display are aligned in the same orientation (i.e., the top of the numbers on both are facing the same direction).
- ☐ State transition diagram is completely specified (i.e., all transitions between states are specified, output conditions specified in each state)
- ☐ Report includes state transition table to document the nextstate and output values for each state based on the current state and inputs.
- ☐ Report explains tradeoffs between the chosen design decisions and alternatives (e.g., why did you select a certain switch debouncing strategy and what are the tradeoffs between your chosen method and others?).

General Specifications

Proficiency

General Schematic Specifications

- ☐ All pin names labeled
- ☐ All pin numbers labeled
- ☐ Crossing wires clearly identified as junction or unconnected
- ☐ Neat layout (e.g., clear organization and spacing)
- ☐ All parts labeled with part number
- ☐ All component values present

Block Diagram

- ☐ Block diagram present with one block per SystemVerilog module
- ☐ Each block includes all input and output signals

HDL & Code Specifications

General Formatting

- ☐ Descriptive filename (e.g., `lab2_jb.sv`)
- ☐ Descriptive variable names
- ☐ Neat formatting (e.g., standard indentation, consistent formatting for variable names (kebab-case/snake_case/camelCase/PascalCase))
- ☐ Descriptive and clear function/module names

Comments

- ☐ Comments to indicate the purpose of each function/module

Lab Writeup/Summary

- ☐ Brief (e.g., 3-5 sentence) description of the main goals of the assignment and what was done.
- ☐ Explanation of design approach. How did you go about designing and implementing the design?
- ☐ Explanation of testing approach. How did you verify your design was behaving as expected?
- ☐ Statement of whether the design meets all the requirements. If not, list the shortcomings.
- ☐ Number of hours spent working on the lab are included.
- ☐ Writeup contains minimal spelling or grammar issues and any errors do not significantly detract from clarity of the writeup.
- ☐ (Optional) List commments or suggestions on what was particularly good about the assignment or what you think needs to change in future versions.

Excellence

General Schematic Specifications

- ☐ Standard symbols used for all components where applicable
- ☐ Signals "flow" from left to right where possible (e.g., inputs on left hand side, outputs on right hand side)
- ☐ Title block with author name, title, and date

HDL & Code Specifications

General Formatting

- ☐ Name, email, and date at the top of every file
- ☐ Comment at the top of each source code file to describe what is in it
- ☐ Clear and organized hierarchy (e.g., deliniation between top level modules and submodules)

Testbenches

- ☐ Testbenches written for each individual module to demonstrate proper operation
- ☐ Testbench output included in the report

Lab Writeup/Summary

- ☐ Writeup is free of spelling and grammar issues

Discussion

A matrix keypad has four rows and four columns, connected to 8 pins, as shown below for two different models. When you press a key, the corresponding row and column are connected. Check this with a multimeter to confirm your pinout.

Mechanical switches are prone to a phenomenon called switch bounce, where the switch makes and breaks its connection repeatedly on a time scale of microseconds to a few milliseconds. You should design your system in such a way that a single keypress registers only once even if there is some momentary bounce.

This is a thinking person's lab. If you thoroughly understand the problem and design a simple scanner circuit, you can complete the lab fairly efficiently. If you go by trial and error, you may find yourself in lab indefinitely. You will need to generate rather complex stimuli to simulate your keypad scanner. Many students have been tempted to skip simulating and debug on the real hardware. Almost all have regretted it. Similarly, some students have been lured down the path of asynchronous design, at their peril.

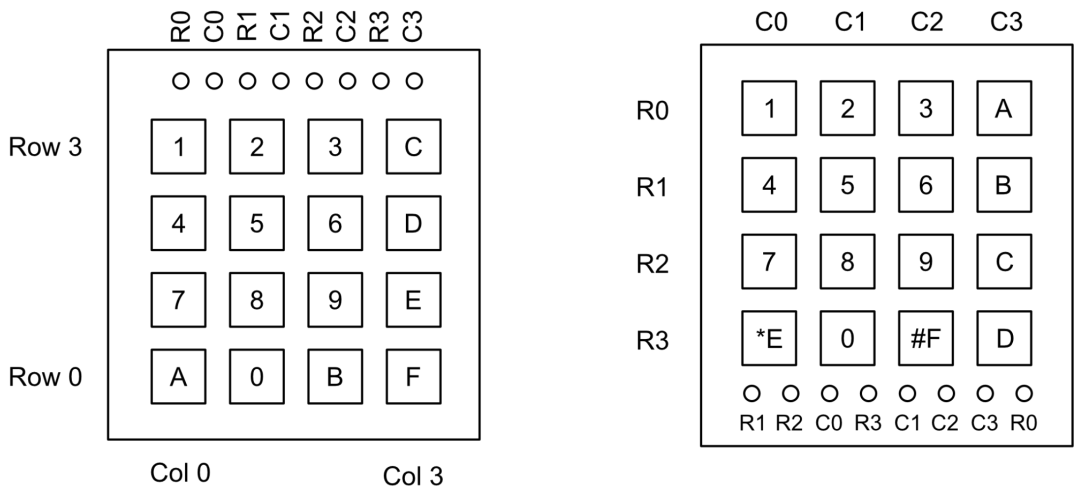

Figure 1: Pinout of keypads.

What to Turn In

When you are done, have your lab checked off by the instructor. You should thoroughly understand how it works and what would happen if any changes were made. Turn in your lab writeup including the following information:

- Schematics of the breadboarded circuit.
- Your Verilog code and simulation results.
- A complete state transition diagram which illustrates the states of your system and the transitions between states. Remember that the transitions from each state should be completely described – in other words, the transitions should cover all possible combinations of the potential inputs.
- How many hours did you spend on the lab? This will not count toward your grade.

Hints

Look at your RTL schematic in your synthesis tool (`Tools -> Netlist Analyzer`). Understand why your code produces the hardware you see. Be sure your combinational logic doesn't have any registers. Be sure your logic has no latches or tristate buffers. The oscilloscope is handy for tracking down timing problems.